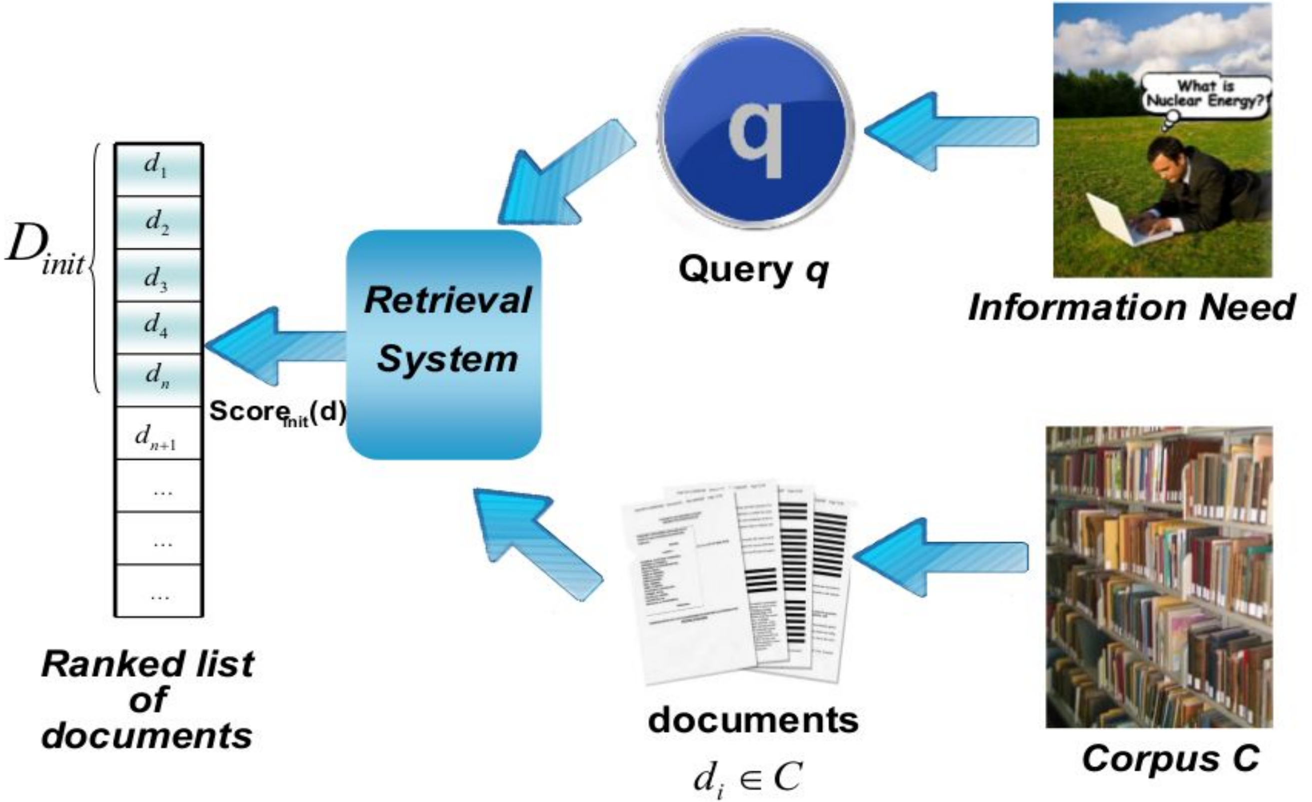
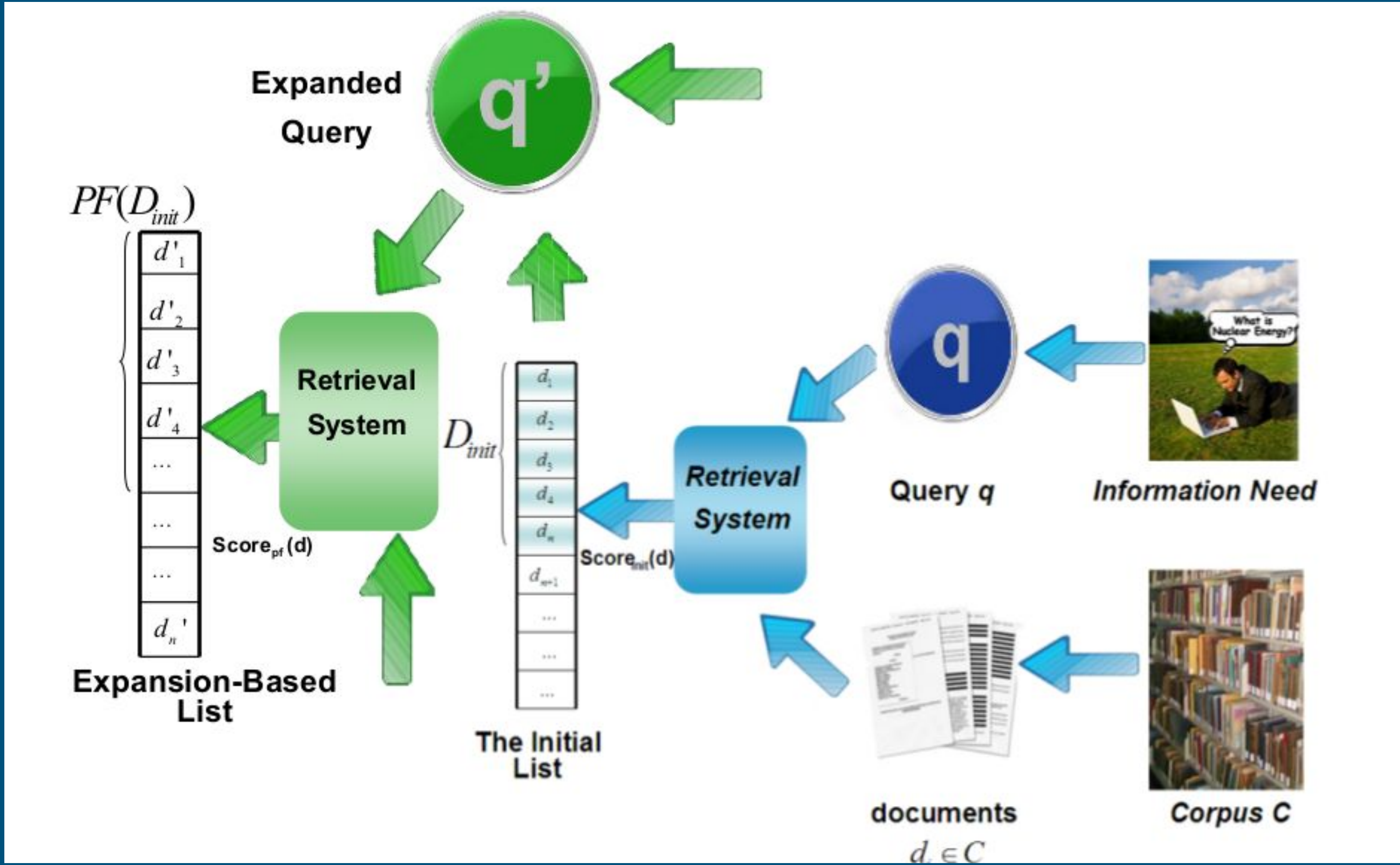
Query-Drift Prevention for Robust Query Expansion

Kunal Suthar
201401131

The initial retrieval process:



Retrieval Process with Pseudo Relevance Feedback



Problem with Pseudo Relevance Feedback

- D init may contain many non relevant documents and the initially retrieved document list D init may not manifest all query-related aspects.
- So, what are the consequences?

Query Drift

- the shift in “intention” from the original query to its expanded form. (Mitra et al. 98’) (e.g., q: "Paris Hilton", q’: "Paris Hilton Whitney model heiress").
- While on average, pseudo-feedback-based query expansion methods improve retrieval effectiveness over that of retrieval using the original query, sometimes the performance is inferior to that of using only the original query.

Query Drift Prevention

- Improving Robustness of a Query using Fusion

Query Drift Prevention

- Improving Robustness of a Query using Fusion
- What do we mean by “Robustness of a Query” ?

Query Drift Prevention

- Improving Robustness of a Query using Fusion
- What do we mean by “Robustness of a Query” ?

Robustness of a query refers to the MAP performance pertaining to that query.

Improving Robustness using Fusion

- ❖ Data fusion - combining retrieval methods or query representations.
- ❖ Why Data Fusion?

Improving Robustness Using Fusion - Motivation

- Using a variety of methods (results) will utilize different aspects of the search space and hence will return more relevant results.
- Documents ranked high by both retrieved lists are potentially relevant since they constitute a good match to both forms of the presumed information need.
- Query expansion can add aspects that were not in the original query but may be relevant to the information need and may improve the retrieval.
- A document that is ranked high by both the initial retrieval and the expansion is assumed (potentially) to suffer less from query drift.

Improving Robustness Using Fusion(Algorithms)

The following retrieval methods operate on D init ∪ PF (D init) :

- The **combMNZ** method rewards documents that are ranked high in both D init and PF (D init):

$$Score_{combMNZ}(d|q) \stackrel{def}{=} (\delta[d \in \mathcal{D}_{\text{init}}] + \delta[d \in PF(\mathcal{D}_{\text{init}})]) \cdot$$
$$\left(\frac{\delta[d \in \mathcal{D}_{\text{init}}]\, Score_{\text{init}}(d|q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Score_{\text{init}}(d'|q)} + \frac{\delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{pf}}(d|q)}{\sum_{d' \in PF(\mathcal{D}_{\text{init}})} Score_{\text{pf}}(d'|q)}\right).$$

For statement s, $\delta[s] = 1$ if s is true and 0 otherwise.

- Note that a document that belongs to only one of the two lists (D init and PF(D init)) can still be among the highest ranked documents

Improving Robustness Using Fusion(Algorithms)

- The **interpolation** algorithm: Differentially weights the initial score and the pseudo-feedback-based score using an interpolation parameter λ :

$$Score_{interpolation}(d|q) \stackrel{def}{=} \frac{\lambda\delta[d \in \mathcal{D}_{\text{init}}]\, Score_{\text{init}}(d|q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Score_{\text{init}}(d'|q)} + \frac{(1-\lambda)\delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{pf}}(d|q)}{\sum_{d' \in PF(\mathcal{D}_{\text{init}})} Score_{\text{pf}}(d'|q)}.$$

Improving Robustness Using Fusion(Algorithms)

- The **rerank** method re-orders the (top) pseudo-feedback-based retrieval results by the initial scores of documents.

$$Score_{re-rank}(d|q) \stackrel{def}{=} \delta[d \in PF(\mathcal{D}_{\text{init}})]\, Score_{\text{init}}(d|q).$$

Evaluation

- **Evaluation methods:**

❖ MAP - Mean Average Precision - effectiveness measurement.

❖ <Init - Percentage of queries for which the expansion-based performance is worse than that of using the original query(measure of robustness).

❖ The implementation of all the previously mentioned fusion based algorithms were done on Terrier 3.6 search engine.

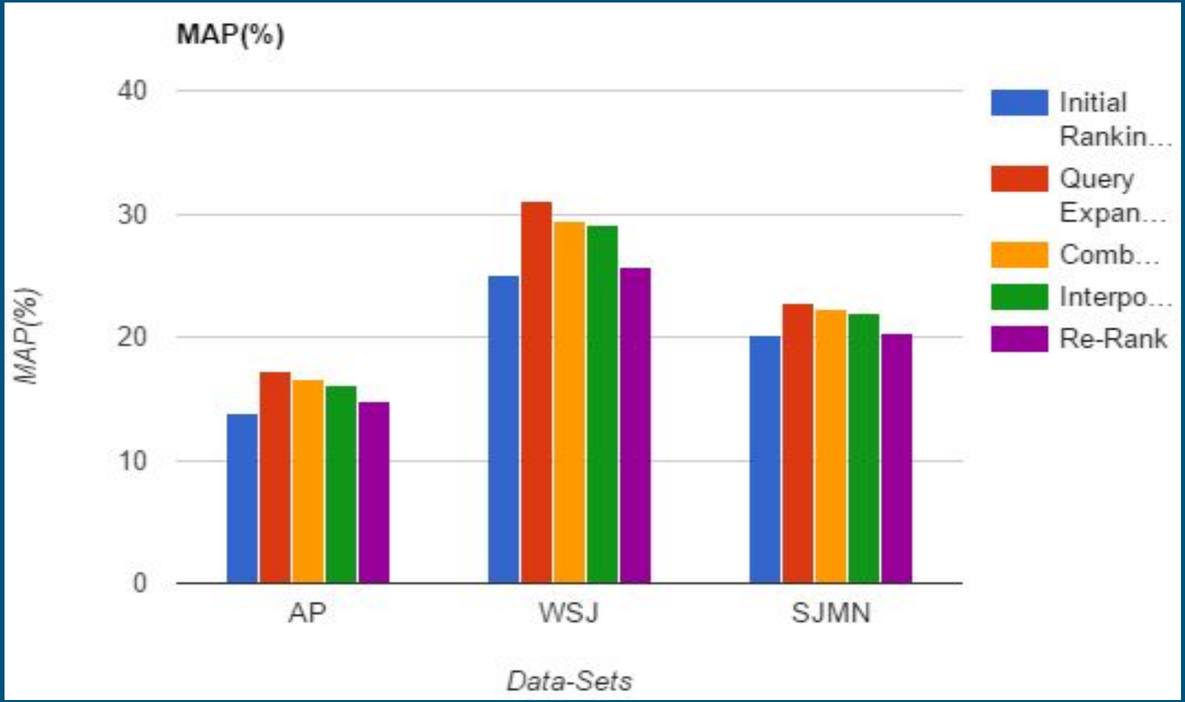
TREC Collections:

Corpus	Queries	Disks
TREC1-3	51-200	1-3
ROBUST	301-450, 601-700	4,5
WSJ	151-200	1,2
SJMN	51-150	3
AP	51-150	1-3

Result Table

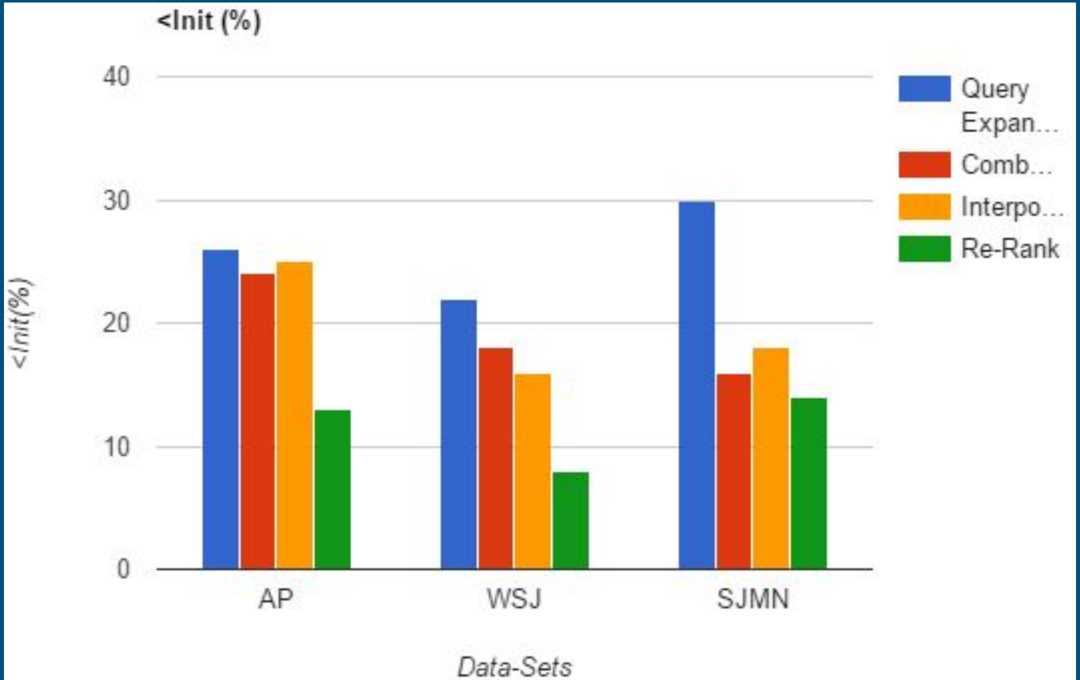
	AP DATASET		WSJ DATASET		SJMN DATASET	
Algorithm	MAP (%)	< Init (%)	MAP (%)	< Init (%)	MAP (%)	< Init(%)
Initial Ranking	13.89	--	25.00	--	20.19	--
Terrier's Bo1 Model	**17.25**	26.0	**31.00**	22.0	**22.76**	30.0
combMNZ	16.51	24.0	29.47	18.0	22.28	16.0
interpolation	16.08	25.0	29.06	16.0	22.01	18.0
Re-rank	14.79	**13.0**	25.65	**8.0**	20.37	**14.002**

Results: Performance in MAP terms



- all fusion-based methods yield MAP performance that is better than that of the initial ranking that utilizes only the original query.

Results: Measure of Robustness(<Init %)



- It can be observed that all the three fusion methods give more robustness(lesser <Init %) than query expansion solely.
- <Init % is calculated by the percentage of queries for which the MAP performance is worse than that of the initial ranking. Lower <init % correspond to improved robustness.

References:

1. Query Drift Prevention for Robust Query Expansion by Liron Zighelnic and Oren Kurland.
2. http://terrier.org/docs/v3.6/ for learning functionalities of terrier.